IN THE SPECIFICATION

Please replace the paragraph beginning at page 6, line 21, with the following rewritten paragraph:

Here, the MAP decoding method will be described taking an example of the foregoing turbo-code with the coding rate of 1/3 and the constraint length of three. Fig. 15 is a block diagram showing a configuration of a conventional decoding unit of the turbo-code. In Fig. 15, the reference numeral 201A designates a ~~decoder~~ decoder part for generating an external value Le from channel values Xl and Y1 and a prior value La according to the MAP decoding method; 201B designates a ~~decoder~~ decoding part for generating an external value Le* and a posterior value L* from the channel value X2 (= X1*) generated by interleaving the channel value X1, the channel value Y2 and the prior value La* according to the MAP decoding method; 202A designates an interleaver for generating prior values La^*_k by rearranging the bits Le_k of the external value Le in accordance with a prescribed mapping; 202B designates an interleaver for generating the bit sequence $X^* = \{x^*_k\}$ by rearranging the bits x_k of the channel value Xl in accordance with a prescribed mapping; 203 designates a deinterleaver for carrying out the inverse mapping of the external values Le^*_k; and 204 designates a decision circuit for estimating the value of the information bits in accordance with the plus or minus of the posterior values.

Please replace the paragraph beginning at page 7, line 14, with the following rewritten paragraph:

Figs. 16A and 16B are diagrams each showing an example of paths on a trellis of the ~~decoder~~ decoder parts 201A or 201B of Fig. 15.

Please replace the paragraph beginning at page 7, line 16, with the following rewritten paragraph:

First, the ~~decoder~~ decoding part 201A calculates the posterior value L_k (logarithmic posterior probability ratio) from the channel values Xl and Yl and the prior value La (La = { La_k {k = 0, 1,. . . , N+1) }) by the following Expression (2). The poster or value L_k represents the reliability of the information bit d_k. It takes an increasing positive value with an increase of the probability of the information bit d_k being one, and an increasing negative value with an increase of the probability of the information bit d_k being zero.

Please replace the paragraph beginning at page 7, line 28, with the following rewritten paragraph:

First, the ~~decoder~~ decoder part 201A calculates transition probabilities $\gamma_k(m^*, m)$ (m, m* = 0, 1, 2, 3) at each point of time k by the following Expression (3). The transition probabilities γ_k (m*, m), which correspond to a branch metric of the Viterbi algorithm, represent the probabilities that the states make a transition from the states m* at the point of time k to the states m at the point of time k+l.

Please replace the paragraph beginning at page 8, line 25, with the following rewritten paragraph:

Subsequently, the ~~decoder~~ decoding part 201A sequentially calculates forward path probabilities α_k (m) (m = 0, 1, 2, 3) from k = 0 to k = N+l using the transition probabilities γ_k (m*, m) (m, m* = 0, 1, 2, 3) by the following forward recursive Expression (5), and stores them in the memory not shown. Here, initial values α_0(m) (m = 0, 1, 2, 3) of the forward path probabilities are set by Expression (6).

Please replace the paragraph beginning at page 9, line 23, with the following rewritten paragraph:

Thus, the ~~decoder~~ decoding part 201A calculates the probabilities α_k (m) of all the forward paths. Subsequently, it calculates the probabilities β_k (m) (m = 0, 1, 2, 3) of the reverse paths by the following reverse recursive Expression (8).

Please replace the paragraph beginning at page 10, line 2, with the following rewritten paragraph:

To achieve this, the ~~decoder~~ decoding part 201A reads out the transition probabilities γ_k (m, m*) from the memory, calculates she reverse path probabilities $\beta_k(m)$ from k = N+1 to k by Expression (8), and stores them in the memory. The reverse path initial values $\beta_{N+2}(m)$ (m = 0, 1, 2, 3) are set according to the following Expression (9).

Please replace the paragraph beginning at page 10, line 18, with the following rewritten paragraph:

Subsequently, the ~~decoder~~ decoding part 201A calculates the posterior value L_k in parallel with the calculation of the reverse path probabilities $\beta_k(m)$ according to the following Expression (11).

Please replace the paragraph beginning at page 10, line 22, with the following rewritten paragraph:

In the course of this, the ~~decoder~~ decoding part 201A read out of the memory the reverse path probabilities β_{k+1} (m*), the transition probabilities γ_k (m, m*) and the forward path probabilities α_k (m), and calculates the posterior value L_k of Expression (2) by Expression (11). The denominator of Expression (11) is the sum total of all the state transitions $m \rightarrow m^*$ when the information bit d_k is zero, whereas its numerator is the sum total of all the state transitions $m \rightarrow m^*$ when the information bit d_k is one.

Please replace the paragraph beginning at page 11, line 17, with the following rewritten paragraph:

The ~~decoder~~ decoding part 201A further calculates the external value Le_k by the following Expression (13), and stores it in the memory not shown.

Please replace the paragraph beginning at page 11, line 23, with the following rewritten paragraph:

In this way, the ~~decoder~~ decoding part 201A calculates the external value Le = {Le_0, Le_1, ...Le_{N-2}, Le_{N-1}} and supplies it to the interleaver 202A.

Please replace the paragraph beginning at page 12, line 1, with the following rewritten paragraph:

The interleaver 202A rearranges the order of the elements of the external value Le to generate the prior value La* = {$La^*_k = Le_{INT(k)}$ (k = 0, 1, ..., N-1)} used by the ~~decoder~~ decoding part 201B.

Please replace the paragraph beginning at page 12, line 4, with the following rewritten paragraph:

The ~~decoder~~ decoding part 201B calculates the posterior value L_k* and the external value Le* = { Le^*_0, Le^*_1, . . ., Le^*_{N-2}, Le^*_{N-1}} from the channel values X2 and Y2 and the prior value La* in the same manner as the ~~decoder~~ decoding part 201A does. The external value Le* is supplied to the deinterleaver 203.

Please replace the paragraph beginning at page 12, line 9, with the following rewritten paragraph:

The deinterleaver 203 rearranges the external value Le* according to the prescribed inverse mapping to generate the prior value La = {$La_k = Le^*_{DEINT(k)}$} to be used by the ~~decoder~~ decoding part 201A.

Please replace the paragraph beginning at page 12, line 14, with the following rewritten paragraph:

The turbo-code decoding unit repeats the foregoing process ~~by~~ a plurality of times to improve the accuracy of the posterior values, and supplies the decision circuit 204 with the posterior values L^*_k calculated by the ~~decoder~~ decoding part 201B at the final stage. The decision circuit 204 decides the values of the information bits d_k by the plus or minus of the posterior values L^*_k according to the following Expression (14).

Please replace the paragraph beginning at page 12, line 26, with the following rewritten paragraph:

As described above, the ~~decoder~~ decoding part 201A successively calculates the transition probabilities of the first received code sequence from k = 0 to k = N+1 for respective points of time in parallel with the calculation of the forward path probabilities $\alpha_k(m)$ (step 1), and then reverse path probabilities $\beta_k(m)$ from k = N+2 to k = 1 for the respective points of time in parallel with the calculation of the posterior values L_k and the external values Le_k (step 2), thereby completing the first decoding of the received code sequence. After that, the ~~decoder~~ decoding part 201B carries out similar processing for the second received code sequence (steps 3 and 4) to calculate the posterior values L^*_k and the external values Le^*_k.